ENGINEERING TRIPOS   PART IIA

ELECTRICAL AND INFORMATION SCIENCES TRIPOS   PART I

Friday 8 May 1998   9 to 12

Paper E6

COMPUTING SYSTEMS

Answer not more than **five** *questions.*

All questions carry the same number of marks.

The **approximate** *number of marks allocated to each part of a question is indicated in the right margin.*

(TURN OVER

1 (a) What is meant by a MIMD *parallel architecture?* [2]

(b) Figure 1 shows the structure of two MIMD machines. Explain why smaller MIMD machines tend to be laid out like Machine A, while larger MIMD machines resemble Machine B. [3]

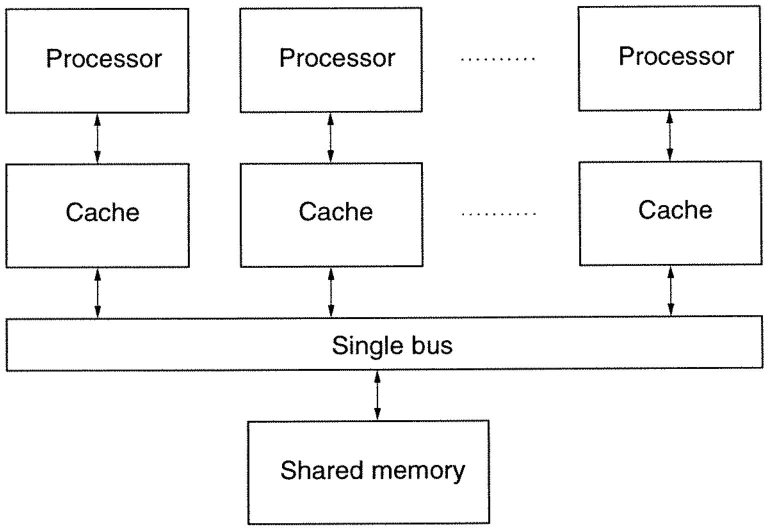
(c) The individual processors in a MIMD machine can communicate through shared memory or by message passing. Is Machine B restricted to either of these alternatives? Justify your answer. [2]

(d) What is meant by *cache coherency*? Briefly describe two ways to enforce cache coherency in Machine A. Assume the caches are capable of *bus snooping.* [4]

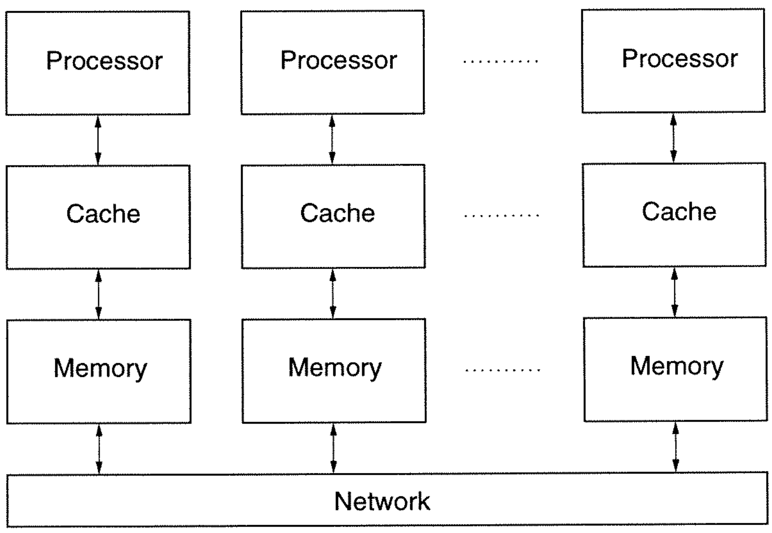
(e) You own a uniprocessor machine, but are considering replacing it with a version of Machine A that contains 10 processors identical to the one in your current machine. You would like your major application to run at least 8 times faster on the new machine. Assume that your application can run in two modes, switching between the two as necessary: parallel with all processors fully used, or serial with only one processor used. What fraction of the application can be serial if your speed-up target is to be met? Comment on the likelihood of achieving this target. [5]

(f) Your major application comprises 10 independent processes which can run in parallel 100% of the time, so you decide to go ahead with the upgrade, anticipating a 10 times speed-up over the uniprocessor machine. You are disappointed to find that the actual speed-up is considerably less than this. Why do you think this is? [4]

(cont.



Machine A



Machine B

Fig. 1

(TURN OVER

2 Discuss the need for, and design of, a virtual memory system. You should structure your answer around the following points.

(a) The requirements of a modern computer system that lead to the adoption of a virtual memory system. [2]

(b) How a virtual memory system satisfies these requirements. [3]

(c) The need for a translation lookaside buffer (TLB). [2]

(d) How a virtual memory system is designed to minimize the amount of disk access, and why this is important. [4]

(e) The factors that influence the selection of a large page size, and also one conflicting factor that demands a smaller page size. [5]

(f) Two alternative ways of ensuring that the TLB does not undermine process protection, and a comparison of the strengths and weaknesses of the two. [4]

3 (a) Explain what is meant by an *abstract datatype* and why the use of abstract datatypes is helpful throughout the production of a large software system. [6]

(b) Illustrate the use of a Pascal module for implementing an abstract datatype by designing such a datatype to represent a pack of cards for use in computer simulations of card games. The operations of taking the top card from the pack, returning a card to the bottom of the pack, and shuffling the pack are to be implemented. A card is to be represented by the following datatypes

```
TYPE suit = (spade, heart, diamond, club);
     value = (two, three, four, five, six, seven, eight, nine,
              ten, jack, queen, king, ace);
     card = RECORD
               s : suit;
               v : value;
              END;
```

and the existence of a random number generator

```
TYPE rand_val = 0..65536;
FUNCTION random : rand_val;
```

may be assumed. You may assume that there will be at most 52 cards in a pack at a time. [8]

(c) Discuss generally the merits of implementing abstract datatypes using Pascal modules or object oriented techniques. Illustrate your answer with reference to the above example taking into account the possibility of the need for additional features to be added for new games at a later stage, for example: more than one pack; packs with certain cards removed; and other "shuffling" strategies such as cutting. [6]

(TURN OVER

4 (a) Explain why there may be problems when several processes in a multi-process system need access to the same data. What criteria must software mechanisms designed to overcome these problems satisfy for reliable operation? Briefly discuss what, if any, special hardware or operating system features are required to implement these mechanisms. [6]

(b) Figure 2 illustrates part of a program which uses doubly-linked lists to provide communication between processes in a multi-process system. The routine `device_ready` determines if there are data available from the input device specified in its argument. `read_device` reads a data item, its operation is undefined if no data are available. `suspend` suspends the current process. `wakeup` restarts the specified process if it is suspended, otherwise it has no effect.

(i) Explain carefully which of the routines `putinlist`, `getfromlist` and `listempty` will not operate reliably and show how these could be made reliable by the use of binary semaphores. Would a monitor provide a better solution? [8]

(ii) Discuss whether the reliable operation of the linked lists using semaphores is sufficient to ensure the reliable operation of the two processes `proc1` and `proc2`. [6]

```
TYPE listptr = ^listcell;
     listcell = RECORD
                        next : listptr;
                        prev : listptr;
                        data : data_type;
                END;
     listhead = listptr;
VAR list1, list2 : listhead;
PROCEDURE putinlist(head:listhead; item:listptr);
BEGIN
        item^.next := head^.prev^.next;
        head^.prev^.next := item;
        item^.prev := head^.prev;
        head^.prev := item
END;
FUNCTION getfromlist(head:listhead) : listptr;
BEGIN
        VAR  p : listptr;
        IF listempty(head) THEN
                p := NIL
```

Fig. 2

(cont.

```
        ELSE BEGIN
                p := head^.next;
                head^.next := head^.next^.next;
                head^.next^.prev := p^.prev;
                p^.next = NIL; p^.prev = NIL
        END;
        getfromlist := p
END;
FUNCTION listempty(head:listhead) : boolean;
BEGIN
        listempty := head^.next = head
END;
PROCESS proc1
BEGIN
        VAR item : listptr;
            data : data_type;
        new(list1); {dummy first item}
        list1^.next := list1;
        list1^.prev := list1;
        WHILE true DO BEGIN
                IF device_ready(1) THEN BEGIN
                        data := read_device(1);
                        <-- process data -->
                        new(item);
                        item^.data := data;
                        putinlist(list1, item);
                        wakeup(proc2)
                END
                ELSE BEGIN
                        IF listempty(list2) THEN
                                suspend;
                        ELSE BEGIN
                                item := getfromlist(list2);
                                <-- process item.data -->
                                dispose(item)
                        END
                END
        END
END;
PROCESS proc2
BEGIN
        VAR item : listptr;
            data : data_type;
        new(list2);
        list2^.next := list2;
        list2^.prev := list2;
        WHILE true DO BEGIN
                IF device_ready(2) THEN BEGIN
                        data := read_device(2);
                        <-- process data -->
                        new(item);
                        item^.data := data;
                        putinlist(list2, item);
                        wakeup(proc1)
                END
                ELSE BEGIN
                        IF listempty(list1) THEN
                                suspend;
                        ELSE BEGIN
                                item := getfromlist(list1);
                                <-- process item.data -->
                                dispose(item)
                        END
                END
        END
END;
```

Fig. 2 (cont.)

(TURN OVER

5 A medical diagnostics problem of identifying an 'adverse' outcome from a 'benign' one involves a single feature variable x. The symbols w_A and w_B are used to denote the adverse and benign outcomes respectively. In this context, Bayes' rule for conditional probabilities is written as

$$p(w_A|x) = \frac{p(x|w_A)p(w_A)}{p(x)}.$$

(a) Briefly explain the various terms in the above equation and how they may be computed from a sample of data. [6]

(b) Assume that the probability density functions $p(x|w_A)$ and $p(x|w_B)$ are both Gaussian, with means of 2.0 and 0.0 respectively. Their variances are both equal to 1.0. Further assume that $p(w_A) = p(w_B)$.

A decision threshold θ is to be set so that a particular outcome is classified as adverse if $x \geq \theta$.

(i) Sketch the two distributions to a common scale and show, using an arbitrary setting of θ, how the different error probabilities can be calculated. [6]

(ii) If, instead, $p(w_A) = 0.1$ and $p(w_B) = 0.9$, starting from Bayes' rule above, derive the optimal value for θ that minimises the total misclassification error. [4]

(iii) Discuss why, in a practical application, a value of θ different from the value calculated above may be used. [4]

6 (a)

(i) Explain what is meant by the *k-Nearest Neighbour* (kNN) rule for pattern classification. [4]

(ii) Show how the kNN rule approximates Bayes' optimal classifier. [4]

(iii) Consider a character recognition task in a system that automatically reads post-codes. Compare the computational and memory requirements of a kNN classifier with those of linear discriminant analysis. [4]

(b) A quadratic discriminant function of a single feature variable x is defined as

$$g(x) = w_0 + w_1 x + w_2 x^2.$$

Deriving any expression necessary to estimate the unknown parameters, explain how this function may be used in a two-class pattern classification problem. [8]

7 (a) State three metaphors used in Artificial Intelligence (AI) to solve problems. Briefly discuss any relationships between them. [3]

(b) Figure 3 shows a simple analogy problem. The correct answer selected by a person is option 2 but a computer programmed using simple AI techniques incorrectly gives the answer 1.

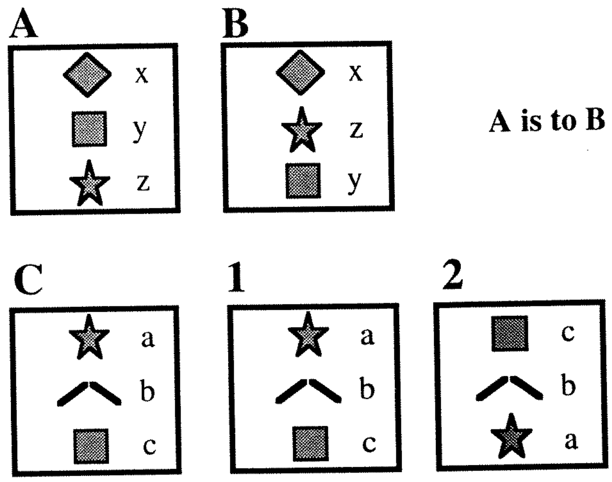
Explain why the computer gives this answer by:

(i) Drawing the rule (semantic) net descriptions for each of the five groups. [4]

(ii) Explaining how these rules cause the computer to reach its conclusion. [4]

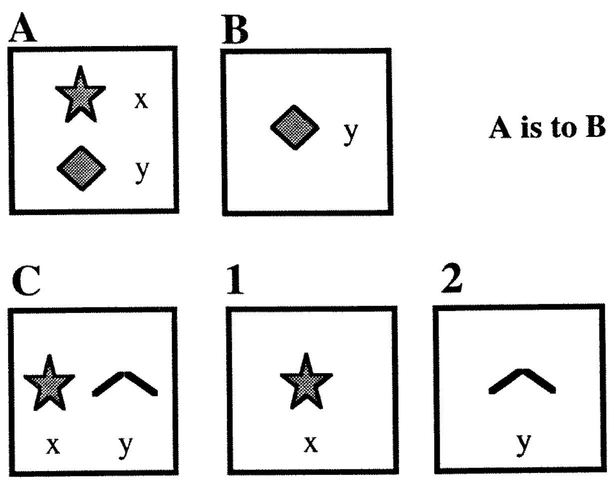
(c) The same type of procedure is applied to the task illustrated in Fig. 4 but the computer is unable to solve the problem. Why might this be so? How could the procedure be enhanced so that it solves the problem correctly? [6]

(d) Comment on some of the practical and other issues that may arise when constructing AI systems to solve these problems. [3]



A is to B as C is to ?

Fig. 3



A is to B as C is to ?

Fig. 4

(TURN OVER

8 Figure 5 shows a diagram of part of a telecommunications network for transmitting message packets. Each packet is sent along a pre-determined route to the goal node (G) calculated by the sending node (S) depending on the prevailing traffic circumstances. You are designing an AI search-based algorithm to calculate the fastest route for each packet before it is transmitted. Initially each packet has a size of 1Kbyte.

The number beside each node in Fig. 5 is the store-and-forward time of the node in milliseconds for a 1Kbyte packet. Assume that inter-node transit time is instantaneous.

(a) State four criteria used to evaluate search strategies. [4]

(b) Show, with the aid of diagrams, how your program would identify a message path using an *exhaustive depth-first* search approach. [6]

(c) If examining a new node in the search tree takes 0.1 seconds, calculate the time required to find the best route using the exhaustive approach. In this example, why is this not a good way to route messages? What might be a better approach? Under what conditions would this method be useful? [4]

(d) The network hardware is now upgraded and after the end-node has received a packet you get a message back showing the route taken and the actual time of arrival at each node.

Describe, with the aid of the diagram that you produced in (b), how you would now improve the search using the A* algorithm. In particular, compare the behaviour of this approach against the exhaustive approach and identify a possible evaluation function. [6]

(cont.

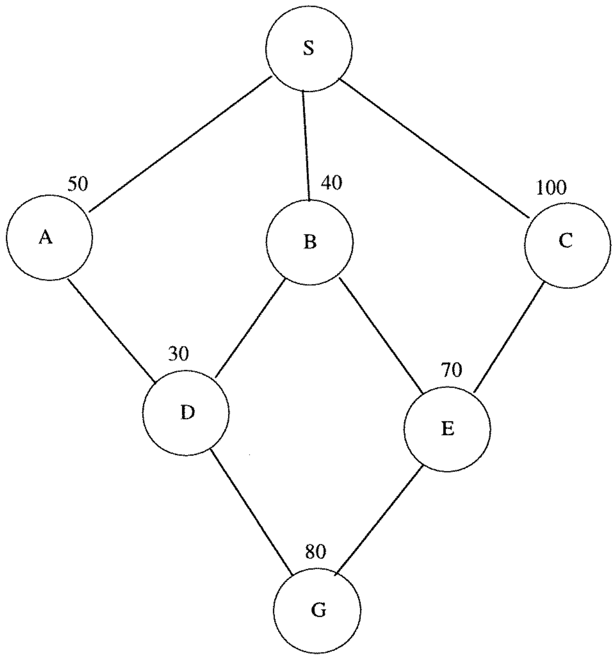


Fig. 5

END OF PAPER